DOCUMENT RESUME

ED 234 755 IR 010 821

AUTHOR Dixon, Robert C.; Clapp, Elizabeth J.
TITLE A Theory-Based Computer Tutorial Model.
INSTITUTION Illinois Univ., Urbana. Computer-Based Education Research Lab.
REPORT NO CERL-Report-E-26
PUB DATE May 83
NOTE 41p.; For related document, see IR 010 820.
PUB TYPE Viewpoints (120)

EDRS PRICE MF01/PC02 Plus Postage.
DESCRIPTORS *Computer Assisted Instruction; *Computer Programs; *Design Requirements; Feedback; Flow Charts; Individualized Instruction; *Instructional Design; *Models; Specifications; Theories
IDENTIFIERS *Corrective Feedback Paradigm; *Tutorial Mode

ABSTRACT

Because of the need for models to illustrate some possible answers to practical courseware development questions, a specific, three-section model incorporating the Corrective Feedback Paradigm (PCP) is advanced for applying theory to courseware. The model is reconstructed feature-by-feature against a framework of a hypothetical, one-to-one, teacher-directed tutorial, indicating the following for each feature: (1) its theoretical and empirical bases, if any; (2) some strengths and weaknesses; (3) its specifications; and (4) research considerations. The model is characterized as being theory-based and sufficiently specific to generate actual lesson designs as well as for research purposes. Instructional features discussed are the pretest, frame, and modeling. Analyses of features related to student and computer/teacher interactions include those concerned with determining the tentative practice set, eliciting a response, judging a response, providing feedback for a correct response, determining error type, revising item type, providing feedback for incorrect responses, and exhausting the item set. Immediate and delayed posttests and lesson revision are also examined, and applications of the model for lesson design are considered. Several figures illustrate the model, and 20 references are listed. (LMM)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

U.S. DEPARTMENT OF EDUCATION
NATIONAL INSTITUTE OF EDUCATION
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

- This document has been reproduced as received from the person or organization originating it.
- Minor changes have been made to improve reproduction quality.
- Points of view or opinions stated in this document do not necessarily represent official NIE position or policy.

Computer-based Education Research Laboratory

CERL

ED234755

A THEORY-BASED COMPUTER TUTORIAL MODEL

ROBERT C. DIXON

ELIZABETH J. CLAPP

CERL REPORT E-26

MAY 1983

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY
D. Bitzer

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

LR010231

University of Illinois at Urbana-Champaign

Copyright © May 1983
by the Board of Trustees, University of Illinois

All rights reserved. No part of this book may be reproduced in any form or by any means without permission in writing from the authors.

The PLATO® system is a development of the University of Illinois.

PLATO® is a service mark of Control Data Corporation.

Acknowledgement

We are indebted to Martin A. Siegel, Director of the PLATO Curriculum Project (PCP), University of Illinois, and to our colleagues on the project. It is the work of PCP which provided the basis and inspiration for this paper.

In particular, it should be noted that the Corrective Feedback Paradigm (CFP), a theoretical concept (Siegel and Misselt, 1983) which has received practical application in numerous lessons, forms the heart of the tutorial model described in this paper. Without CFP, meaningful application of the model would be impossible.

A THEORY-BASED COMPUTER TUTORIAL MODEL

Robert C. Dixon
Elizabeth J. Clapp

PLATO Education Group
Computer-based Education Research Laboratory
University of Illinois at Urbana-Champaign

Harold Mitzel (1981), Walter Wager (1981-82), and others have brought attention to a critical shortcoming of much computer-assisted instruction (CAI): a lack of the application of instructional theory to the development of courseware. This shortcoming cannot be overemphasized. All courseware designers, certainly, aspire to develop pedagogically sound, effective and efficient courseware. There is some indication, however, of a gap between this universal intention and actual practice. After an exhaustive investigation of computer courseware practice, Francis D. Fisher (1982, p. 19) concluded that "Educational courseware is beginning to appear, but most is simplistic in design and fails to exploit the full teaching power of the computer." Alfred Bork (1981-82, p. 6) posed this disturbing question: "Why is it that our actual practices in education should be so divergent with at least the few things that seem to be agreed upon in the educational process?"

An obvious reason for such a gap may be that a somewhat high percentage of courseware is developed without any intention of applying instructional design theory. The alternative to theory-based development, apparently, is "seat-of-the-pants" development. Mitzel (1981, p. 97) has charged that most teacher and pupil behaviors "...appear to rest primarily on atheoretical propositions, hand-me-down techniques and historical inertia." The same thing might be said for "courseware behavior."

It may also be the case that many courseware developers endorse <u>in theory</u> the notion of theory-based design, but find the rigors of either designing or programming such tutorials too demanding in practice.

Or finally, many developers may be quite willing to endure any such rigor, but find specific theory-based models few and far between. What kinds of lesson features, for example, specifically contribute to the theoretical notion of a fully adaptive computer tutorial? Models are needed to illustrate some possible answers to such practical courseware development questions.

To those who are interested in specific models for applying theory to courseware, we address the remainder of this paper, which advances one of potentially many such models. We will reconstruct the model feature-by-feature against a framework of a hypothetical, one-to-one, teacher-directed tutorial, attempting at each feature to indicate:

1. the theoretical and empirical bases, if any, for each feature of the model.

2. some strengths and weaknesses of each feature, relative to teacher-directed instruction.

3. the specifications for each feature.

4. some research considerations.

Note that the model is not offered dogmatically. First, it is the reflection of theoretical considerations expressed in several theories. Foremost among the design theories influencing our model are those of Bloom (1956), Gagnē (1970), Markle (1978), and Engelmann and Carnine (1982), none of which is articulated specifically for technological application, but all of which are adaptable to such application. Some of the principles reflected in the model are common across theories, and some are unique to a particular theory. Second, we recognize that any particular theory can be expressed in a variety of different models, each, in turn, capable of generating a wide variety of instructional lessons. Our rationale for offering such a model is that it is, in fact, theory-based, and that it is sufficiently specific for generating actual lesson designs, as well as for research purposes.

A Theory-based Tutorial Model

One-to-one, teacher-directed instruction long has been recognized as an instructional mode with great potential for highly efficient and effective instruction on the one hand, but as the most expensive and, therefore, the most impractical instructional mode on the other hand. The goal of our model is to capitalize as greatly as possible on the advantages of one-to-one instruction while circumventing its prohibitive expense.

What is it, exactly, that a good teacher does with a student in a one-to-one instructional setting? Obviously, there is no definitive answer to such a question, because a good teacher will act variably according to the variabilities encountered across the students and the skills being taught. That fact, however, does indicate one of the general characteristics of solid computer-based education (CBE) models: the accommodation of variability. Although we cannot tell what a good teacher would do exactly in an idealized instructional setting, we can create a general scenario and identify its characteristics.

Section I

The first section of our model involves instructional features which, when taken together, can be characterized as "initial teaching." The importance of these initial teaching features in the total picture of "complete" teaching is:

a. the assessment of skills, and

b. the initial presentation of information.

These are not, however, central features of the model, since they do not provide or allow for the sophisticated interaction and high level of adaptation that is possible at later points in the model.

Present pretest. Under optimal conditions, a good teacher with just one student would never think of beginning instruction without determining what instruction that student actually needed.

The good teacher would give some kind of curriculum-based pretest to determine (1) whether a student lacks the requisite preskills assumed by the instruction, and (2) whether the student already has the skills identified in the lesson objectives. Pretests are fairly easy for a teacher to administer to one student at a time. Computers are also quite capable of administering pretests, and, in fact, can potentially do so more efficiently than a teacher can, due to the development of computerized adaptive testing.

Pretesting plays a role in individualizing instruction. While it is quite safe to say that any two students differ across a wide range of dimensions, having said so implies very little for the design of

instruction. The differences among humans that do have implications for designing lessons are related directly to the lesson in question. The most general function of the pretest is to identify students who are similar with respect to their knowledge of what is being tested.

Although the students placed in the lesson will be similar with respect to the skills being taught, they will also undoubtedly differ with respect to specific tasks presented within the lesson. Those differences will be revealed later in the lesson through the students' performance; thus, the primary burden for individualizing the instruction falls upon other features within the lesson, which will be discussed presently. It could be said, then, that the pretest is a "gross" discriminator of individual differences. Lessons developed without pretests have an extremely low potential for individualization, while lessons with pretests have potential, but only potential, for individualization.

Thus, the first feature of our model is pretesting. Students who lack prerequisites are routed to earlier instruction, while students who demonstrate mastery of the object skills are advanced to another pretest.[1]

Frame. The good teacher today is likely to precede instruction with a rationale for studying the upcoming material and an explanation of what will be expected of the student. In theory, this practice gives the student a general orientation toward what is coming, and hopefully, some greater motivation to engage actively in the instruction. Assuming that this portion of the instruction is not unduly time consuming, it is a difficult notion to argue with. On the other hand, empirical evidence for the efficacy of this practice appears to be scant.

We include framing in our model because it may very well be effective; and as long as it is done efficiently, it seems at worst harmless. In our own lessons, we accommodate framing through a combination of an informative lesson index and short statements of purpose, which precede the next feature of the model. The theoretical support for framing comes primarily from Gagnē and Bloom.

Neither a teacher nor a computer has any appreciable superior capability for framing lessons that is apparent to us.

Model. Once it has been determined that the student needs and has the necessary preskills for the instruction at hand, the good teacher is ready "to begin teaching." This phase of the instruction normally takes the forms of demonstrating, telling, showing, or otherwise communicating new information to the student. What the teacher actually does at this point should be dependent upon the the type of learning involved. If a new concept is being introduced, the modeling might consist of a definition, or a definition and examples, or just examples. If a rule is being introduced, the student is often told what the rule is and is shown some instances of its application. Another alternative is to show the examples first, then follow with the rule, which can be given by the

teacher or "discovered" by the student. Gagnē described conditions for eight types of learning; Engelmann and Carnine's theory of instruction explicates very specific details for a somewhat more refined set of learning types.

Yet there is no wide agreement among theorists on the details for this phase of instruction. Although experimental research is being conducted on the details of considerations such as how best to present concepts, rules, or problem-solving strategies, results are inconclusive and often contradictory. Much more research on such questions is clearly in order.

An important aspect of modeling that receives almost unanimous support by theorists and research alike is that models alone do not teach effectively. Simply giving students information in some fashion is not teaching. This does not diminish the importance of modeling, but does help put it in a realistic perspective. In a one-to-one teaching situation, it is difficult to imagine a good teacher simply talking at the student, explaining and demonstrating the target skill or skills. Although this "telling" mode of instruction is common off-line (such as through most books and lectures), its commonness, as Bork pointed out, is not to be confused with effectiveness or desirability. Rosenshine (1982, p. 5) identified six general teaching "functions" characteristic of effective instruction. "Presenting new content/skills" is just one of the six teaching functions identified. Susan Markle (1978) quoted R.F. Mager: "If telling were teaching, we'd all be so smart we could hardly stand it."

We might add that a great deal of computer-delivered instruction is comprised either largely or entirely of one-way, linear modeling. In its worst form, such computer instruction is commonly referred to as "page turners." Unfortunately, many such computer lessons are also referred to as "tutorials."

As we have characterized modeling thus far, it is not interactive, and need not be, if upcoming features are adequately interactive. There are circumstances, however, in which the modeling phase of a lesson ought to be interactive. Most notable among those are lessons which teach skills involving many steps, such as long verbal chains, algorithms, and cognitive routines. In theory, at least, modeling for such lessons should include checks of student understanding at each step of the modeling. Also, if the modeling for a particular lesson necessarily must be somewhat long, questions about what is being modeled are appropriate to prevent inattentive students from pressing RETURN or another key until they reach a part of the lesson where they must pay attention.

Figure 1 illustrates the features of the model we have discussed thus far. Provision is made for skipping the modeling phase of instruction under certain conditions. The first condition occurs when the skill being learned is a set of multiple paired associates or verbal chains. Imagine, for instance, that a teacher is giving one-to-one instruction on Spanish vocabulary. Because there is little, if any, potential for

generalization from one vocabulary pair to another, the implication for modeling is that all pairs must be modeled. But if there is a chance that the student might know some pairs in the set, then the modeling for those pairs unnecessarily wastes time. A more efficient approach in this instance is to assess the student's knowledge of the pairs within the lesson, providing instruction on only those pairs the student does not know.

Another condition under which the model may be skipped occurs when the objective of the lesson includes having the student discover the information that would otherwise be presented during modeling.

Insert Figure 1 about here

Teachers have some advantages over the computer relative to modeling, and vice versa; e.g., the teacher creates examples "on the spot" that are likely to be particularly meaningful for a particular student. For instance, a teacher may say to a student: "Remember how Toby felt the other day when you won the citizenship award? That's envy." (Note that this advantage is likely to diminish for group instruction, where it becomes increasingly difficult to create examples that have special meaning for each and every student.)

The computer, on the other hand, is capable of delivering dynamic modeling, which could be difficult, expensive, or impossible for a teacher to deliver, even in one-to-one instruction. Physics concepts, such as velocity and acceleration, are modeled more effectively by dynamic presentations than static ones, as are minimal differences between many concrete concepts.

This portion of the tutorial model is similar to the beginning of a model proposed by Walter Wager (1981-82, p. 262), based upon the instructional design theory of Gagné. (See figure 2.)

Insert Figure 2 about here

Section II

The features in this section of the model involve interactions between student and computer/teacher. Whether or not the instruction is truly individualized and mastery-based is dependent upon these features.

Enter tentative practice set. In most one-to-one teaching situations we can envision, an observer would see the student doing things during most of the tutorial session. Even meaningful, covert cognition can be evaluated only through overt responding. If the student is learning to solve problems, a fair portion of time, presumably, will be spent solving problems. The student who is learning concepts will be

identifying instances of those concepts and discriminating each concept from the others. The student who is learning verbal associations, such as foreign language vocabulary or basic math facts, will be working on making those associations. Practice in applying even the loftiest of the "higher" cognitive skills is generally assumed to be necessary for all but the very rarest of students.

The teacher who is working one-to-one with a student knows that each student will require different amounts and types of practice before achieving mastery of the targeted skill. However, the teacher probably cannot do much more than guess which students will need what amounts and types of practice before practice actually begins. These specifics are revealed most precisely once practice begins.

But practice must begin with some preconceived set of items, which can be modified as necessary once practice commences and the teacher begins to receive precise information from the student's performance. There is a variety of ways of developing an initial set of practice examples. One method we have found to be analytically sound and practical has the designer pose this question: if the student performs without making a single error, how much practice and what kind of practice would I accept as indication that the student has learned what I intended?

Let's assume, for instance, that the teacher is teaching the poetic concept set "pastoral." This teacher is not certain which students will learn the concept quickly, having practiced only a few discriminations, and which will require a great deal of practice with accompanying attention from the teacher. Using the question above as a guide, focusing on errorless performance, the teacher in this situation might think, "If a student could identify two classical pastorals, such as one by Theocritus and one by Virgil; and two from the British Romantics, such as Wordsworth and Coleridge; and two modern pastorals; and could discriminate those from two non-pastorals each from about the same periods and of similar form, all without making an error, I would be satisfied that the student had, in fact, successfully mastered the concept pastoral."

This method for determining the members of a tentative practice set is necessarily subjective, since it is based upon the goals and objectives of the instruction, which, in turn, are subjective. One teacher may feel that performance on three examples of pastorals from each period is a better indication of understanding. Another may feel the two examples per period are adequate, but believe that works by other poets would be more representative. And yet another teacher may feel that the concept pastoral simply is not important for students to learn.

In actual practice, teachers or instructional designers are frequently able to side-step the question of what ought to be taught by deferring to outside authorities: school district or adoption committee objectives, guidelines established by the National Council of Teachers of English, respected theorists, and so on.

If the objectives of the lesson include discovering a concept, a somewhat different question is posed, but the answer is likely to be about the same: from what set of examples is the student most likely to correctly induce the critical features of the concept without either overgeneralizing or undergeneralizing?

Tiemann and Markle (1978) offered clear guidelines for determining the exact number of positive and negative instances necessary for exhaustively demonstrating concepts with a determined number of critical and variable features--a set of examples they referred to as a "minimal rational set." However, the number of minimal rational sets to use for determining that a student has mastered a concept is still a largely subjective matter and requires further research.

When skills involving a series of steps are being taught, "one" practice item consists, in reality, of all the steps required by the skill. In this case, the practice set comprises a number of subsets, each of which requires eliciting several responses.

Our model assumes that someone, somehow, has a very clear conception of what the instruction is supposed to accomplish for the student, since even rudimentary tasks (e.g., developing a tentative item set) are dependent upon such an assumption. All instructional design theories we are aware of support this assumption.

Immediately following modeling (or in some cases, framing), the student begins practicing the new skill on a set of items that is fixed only if the student performs flawlessly.

Computer tutorials have no particular advantage over teacher-directed tutorials at this point, nor do teacher-directed tutorials have any particular advantage over the computer. In either case, the tentative practice items have to be "stored" somewhere. In either case, a human initially has to develop the item set.

Elicit a response. Students work on a set of items so that responses can be elicited. Responses from students provide a computer tutorial with something to react to. Essentially, each response functions as a formative evaluation. In turn, according to Block (1971, p. 79) "Formative evaluation provides the information necessary to individualize instruction within a mastery strategy." Our good teacher, delivering one-to-one instruction, will presumably begin adapting that instruction the moment he or she begins to receive information from one instance of the student's performance. From the practice set, the student receives the type and amount of practice necessary for retaining and transferring new skills. The teacher, on the other hand, receives the types of information from the student necessary for delivering fully adaptive, individualized instruction.

The exact details of response conventions are dependent upon the performance objectives of the lesson, the nature of the skill, and how the skill relates to other skills being taught. Engelmann and Carnine

provide many varied examples of tailoring response conventions to meet these and other criteria, as does Markle.

Computers and teachers are about equally capable of eliciting responses, although teachers have a clear advantage when it comes to judging those responses.

Judge a response. Initially, we may think of responses as either right or wrong, given that later phases of the model accommodate the strong possibility of many distinctive types of wrong responses, and depending upon the skill being taught, a full set of correct responses. It is also possible in many cases for part of a response to be correct and part to be wrong.

The sophistication of judging is a function of the nature of the task, the response conventions called for in the task, and the possible range of student performance, right or wrong, on the task. The extent to which judging should be considered sophisticated ought to depend upon instructional design considerations, not programming considerations.

It is often appropriate, for instance, to ask "yes/no" questions as part of an initial teaching presentation for a concept that is not yet being discriminated from other known concepts. The nature of the task is that of gross discrimination: yes, the instance is an example of the new concept, or no, it is not. The response conventions for such tasks, however, can vary considerably. One possibility is simply multiple-choice: y. = yes, n. = no. On the other extreme, a yes/no question may be asked, but the student is allowed to make an open-ended response. Judging for the latter task form that allows for the range of possible responses (you betcha, I seriously doubt it, sure, hardly, etc.) is programming-sophisticated judging: the programming for such a task is far more sophisticated than for the two-choice task, but for no good reason. Neither task is more meaningful than the other, but in computer applications, the two-choice task is far more efficient and economical.

Programming-sophisticated judging becomes a design problem when it is applied to a task that is inappropriate in the first place. Assume that some students, who recognize instances of the concept "triangle" are about to learn their first specific type of triangle: right triangles. Initially, a yes/no task is appropriate because the discrimination being applied is binary. The new concept is being discriminated only against one "old" concept: triangles. Later in the instruction, after students have learned not only right triangles, but isoceles triangles, equilateral triangles, and so on as well, a different task type is appropriate because each concept is being discriminated against all the others. At that point in instruction, the most appropriate task would involve showing students instances of the various concepts and asking what each is. If a yes/no task with unnecessarily sophisticated judging is used in this latter case, then not only are resources being wasted, but they are being wasted on a task that is instructionally inappropriate as well.

"Judgments about judging" should be made with respect to the appropriateness of the task form in the first place. The "What is this?" task with open-ended responding provides both the opportunity and the necessity for relatively sophisticated judging, which is not a waste of resources when more information is gleaned from the task than would be possible with a choice-response form for the same task type. For instance, production-response formats clearly yield more information on a student's understanding of concept sets than do multiple-choice formats. Production-response formats are undoubtedly more difficult for a computer to judge, but that is not a reason for avoiding such formats when they are appropriate.

There is a reasonable limit, of course, to the computer's capability to judge open-ended, production responses. No widely available technology exists for judging students' oral responses, which teachers, of course, judge routinely and easily. To the extent that skills, such as decoding or foreign language pronunciation, are taught on the computer, severe compromises are usually required relative to what a good teacher can do. Similarly, the kinds of computing resources required even to attempt to judge a spontaneously written composition are enough to discourage most of us from trying to push computer tutorials too far. (This is not to say, in Robert Taylor's (1980) terms, that the computer can't be used as a writing "tool." It just suggests that this particular tool for writing is not necessarily the ideal tutor for that skill area.)

The capability of judging the most common formats in which a given skill will ultimately be used is a reliable criterion for determining whether a computer tutorial is an appropriate mode for delivering instruction in that skill.

Every response elicited from the student is evaluated in this section of the model, in part, as the means of providing appropriate feedback to the student, and, in part, as the basis for adjusting further instruction to fit the needs of the student.

Provide feedback for correct response. The possibilities of feedback for correct responses differ enough qualitatively and quantitatively from those for wrong responses that we think of them as a separate category. Feedback for correct responses can simply (and sometimes justifiably) give knowledge that the response was correct.

Frequently, feedback given for correct responses is motivational. The evidence is weak, however, that such feedback for every response is demonstrably reinforcing, even if it is randomly varied, or is particularly clever, as is the case with some animations. On the other hand, such feedbacks, at worst, probably are harmless, assuming they don't cumulatively amount to a significant loss of time.

A third category of feedback for correct responses is instructional feedback, which may provide a reason why the response is correct, give information that will help discriminate that response from upcoming responses, indicate other acceptable responses, or help make connections

with upcoming or previous instruction. In a lesson on classification, for example, the student might correctly identify "guitar" as a type of musical instrument. The feedback could anticipate the introduction of the concept "class" by saying something like, "Yes, guitars are in the class of musical instruments." When instructional feedbacks are very elaborate, that is a fair indication of some inadequacy in the initial teaching presentation.

Knowledge of response feedback, in particular, in some instances, may be "intrinsic" (Friend & Milojkovic, 1982); that is, the correctness of a student's response can sometimes be made immediately apparent to the student as a natural outcome of performing the task. A very good example of capitalizing upon the potential for intrinsic feedback is a lesson called "Darts" by Sharon Dugdale and David Kibbey (1978). As the screen displays in Figure 3 indicate, the object of the task is for the student to break balloons situated along a number line (which changes at regular intervals). When the student makes a correct response, a balloon breaks. External (or extrinsic) feedback, such as "Good. You broke that balloon," is completely unnecessary for a task such as this, and very likely is far weaker than the intrinsic feedback of seeing the balloon broken.

Insert Figure 3 about here

Intrinsic feedback is often possible for many kinds of math lessons, as Dugdale has repeatedly demonstrated, as well as many "math-related" lessons: physics, chemistry, mechanics. Also, the kind of feedback provided by simulations is intrinsic. Feedback such as this is difficult for a teacher to deliver directly. In addition, many subject areas or task types do not lend themselves to intrinsic feedback. However, when intrinsic feedback can be given, it probably should be.

When a practice item requires doing a sequence of steps, such as "working out" a long division problem, all the required responses must be correct in order for the whole item to be considered correct.

Our good teacher is more capable of delivering "higher quality" feedback for correct responses than a computer tutorial. For instance, although both teacher and computer can space motivational feedbacks to avoid saturating the student with them, only the teacher can do so on the basis of observations of the student doing things such as yawning, looking around the room, and so on. Only the teacher can observe signs of "struggling" in the student's face, a sign that stronger feedback than usual may be appropriate. These advantages of the teacher are by no means overwhelming, but are advantages nonetheless.

Providing correct response feedback is the easiest feature of the model to implement: no courseware that we have seen has had any particular difficulty dealing effectively with students who never make an error. In fact, it might be said that in many cases, this feature is too easy

to implement. That is, it is too easy to construct tasks that students may get right for the wrong reasons. Susan Markle provides numerous examples of such tasks (as well as numerous examples of tasks in which correct responses may be made only for the right reasons).

Determine error type. Although all the students who get a particular item exactly right certainly differ in many respects, they do not differ across any dimension that is significant relative to their performance on that item. That is to say, the individual differences among students, which individualized instruction seeks to accommodate, are differences that are revealed when students perform on particular sets of skills. Those differences are not revealed by perfect performance: all students who perform perfectly on a set of tasks are the same with respect to that set. Significant differences among individual students appear when students perform differently on a set of tasks. There may be a few different ways to perform perfectly, but there is a nearly infinite variety of ways to perform imperfectly. This variety is reflected in the variety of error types students may produce within a given task or set of tasks. The determination of error type, therefore, is a most critical consideration in the individualization of instruction, and is, therefore, an important feature of the model.

Certainly one trait of a good teacher is sensitivity to the types of errors that students make. Ability to determine the implications of each type of error and adaptation of subsequent instruction based upon those implications are also indications that the teacher in question knows and is concerned about what the student is doing.

A substantial number of distinct error types is possible, depending upon several factors: Is the error chronic? Might the error be caused primarily by carelessness rather than by a cognitive difficulty? Could a memory lapse rather than any sort of confusion account for the error? Is this simply an error in following directions--a response-convention error? Is this error indicative of a discrimination problem or a problem producing a response, or possibly both? Could the error be caused by a lapse on an assumed preskill, rather than by a difficulty with the skill that is being targeted?

Truly individualized instruction will note the different qualities of "wrongness" in different error types and subsequently will adapt to the particular difficulty noted at this point in the model.

When skill application tasks are constructed to correspond with the most usual or most useful application of that skill, great varieties of error types are usually within the realm of possibility. CAI lessons sometimes "hide" the fact that possible error types are not being determined by resorting to simple (simple to design and program) response conventions, such as multiple choice, in which every distractor is plain wrong. For instance, a typical application of capitalization rules in off-line instruction asks students to proofread sentences such as:

dr. James' sister, Mrs. douglas, lives in seattle,
but her brother lives in Santa monica.

The computerized version of such a task, however, is often something like this:

Press the letter for the sentence that is capitalized correctly:

1. Dr. James lives in Santa monica.
2. everyone likes dr. James.
3. Professor Smith lives in Seattle.
4. mrs. Douglas visits Florida every summer.

Numerous error types are possible for the first task. The student may fail to capitalize a word or several words that need capitalization, or may capitalize something that does not need it, or may uncapitalize something that is already capitalized correctly, or any combination of these possibilities. Computers can accommodate such a broad range of possibilities if they are instructed to do so.

The second task--multiple choice--allows for only a limited range of error types, which means the task format is capable of rendering only limited information about what the student knows and does not know, which, in turn, limits the potential of the lesson for delivering individualized instruction.

A unique type of "error" is a correct response that is not made within a set period of time. Rapid recall is an appropriate objective for many classes of tasks, such as math facts and vocabulary. To make this judgment, of course, the teacher (or lesson) must check for responding time.

A good teacher, presumably, would not construct simple, artificial task formats simply because they are easy to correct. Rather, a good teacher is likely to think ahead to the kinds of applications for which the skill will most likely be used, and will construct practice tasks accordingly. Sometimes, a multiple-choice task, for instance, is appropriate for meaningfully practicing and assessing a given skill. In other cases, multiple-choice task formats not only yield little information about what a student knows, but also often give the student little practice applying a new cognitive skill in a useful manner. In short, tasks should be designed so that they are compatible with the ultimate application of the skill, and the judging for such tasks should be sensitive to the variety of error types possible within a carefully designed task.

The next two features are the most critical for response-sensitive, adaptive instruction in our model. Many models for adaptive computer tutorials readjust the instruction based upon a student's performance on a set of practice items. The readjustment of instruction embodied in our

model is based upon a student's performance on each practice item encountered, since various types of performance on each item may imply different student needs.

Some models, in fact, do readjust instruction on the basis of individual items, but only make single adjustments, such as lengthening the item set or rearranging the item set. Our model suggests making several simultaneous adjustments in the instruction based upon each student response.

Revise item set. Some revision of the item set is implied by any conceivable type of error, other than for errors in following directions (response convention errors). There are basically two types of set revision. The first is quantitative: the number of items in the set and their location. The second is qualitative: the types of items added, and whether or not discrimination training is required.

In quantitative revision, an error on a single item implies the need for more practice on that item or item type. The application of this principle is one of the clear indications that individualized instruction, in fact, is being delivered: students whose performance indicates they need more practice get more practice. Additional practice serves several important functions: the enhancement of retention, better understanding of the skill, and continuous assessment to insure that the desired retention and understanding are taking place (or to discover that they are not taking place, and that even further modification therefore is required).

In addition to simply providing the student with additional practice, it is important, for the sake of retention, to consider carefully exactly where that additional practice will take place. There is strong empirical support indicating that the additional practice should be spaced across time according to an increasing ratio review schedule. For instance, the item might be placed in a short-term review position (such as the second item after the missed item), a medium-term position, and a longer-term position. This method of item set revising is designed to shape retention and understanding over increasingly greater periods (Siegel & Misselt, 1983).

One type of qualitative modification is indicated whenever the skills being taught are generalizable. For instance, if the student is learning a rule for changing the letter "y" to an "i" when adding a suffix to words, and that student misspells the item "carried" as "carryed," then the additional items should be different words, but of the same type, such as "married" or "studied." To provide additional practice in this situation on just the item that was missed originally is to encourage that the missed item be learned by rote, an outcome quite antithetical to the goals of rule-learning. A change in the exact item, conversely, makes memorization of generalizable items almost impossible, and insures that the generalization is being learned. CAI lessons typically fail miserably to exploit this relatively apparent principle of learning and clear capability of computers, possibly because doing so is no trivial programming matter.

When a nongeneralizable, paired-associate item, such as "lobo --> wolf" is being taught, that exact association, in the direction(s) it is being taught, should be the instance of additional practice.

Another type of qualitative set modification is indicated for sets of both generalizable and nongeneralizable skills whenever an error implies confusion within a set. This modification of the set gives a student discrimination practice when the student's response indicates a discrimination difficulty. This type of lesson modification takes place in the Park and Tennyson (1980) model for teaching coordinate concepts. However, since discrimination is an important factor in many classes of learning besides concept learning, we believe this type of lesson modification is warranted for any type of discrimination error.

Consider, for instance, a paired-associate task in which students are required to identify bones of the body. An arrow points to the mandible on a graphic, and the student is asked to identify that bone. The student who says "mandella" is not confusing the mandible with another bone, but instead appears to have only a partial memory of the association. In this case, any other item in the set of bones being learned is appropriate as a next item. But the student who says "maxilla" is confusing two bones that are located near one another and have similar sounding names. In this latter case, it is appropriate to revise the item set so that mandible and maxilla are discriminated successfully before moving on to other associations. This procedure prevents an obvious confusion from getting worse through practice with intervening items that cannot clarify the confusion, and which additionally confound the confusion by making retention of the feedback more difficult.

A response-sensitive method for providing discrimination training is similarly important for teaching coordinate or similar sets of generalizable skills, such as concepts and rule applications. As with association tasks, a discrimination error on a generalizable skill indicates confusion not only of the skill represented by the missed item, but of the skill represented by the student's response as well (Tiemann & Markle, 1978). If, for instance, a right triangle is presented and a student identifies it as an isoceles triangle, that student is demonstrating a lack of mastery of both concepts. The immediate item set revision implied for such an error is the juxtaposition of instances of isoceles and right triangles.

Teachers certainly may revise items sets, and good teachers probably do in one-to-one instruction. Computers, however, have some advantage over teachers with respect to this feature. If systematic increasing ratio review is provided for a student every time an error is made, the exact positions of review items within the ever-changing set would probably be too much for any human teacher to remember accurately. The computer, however, has no difficulty with the implementation of this feature.

Figure 4 shows a fairly simple error pattern that a computer can manage easily, but which would challenge the memory of most humans. The

original, tentative item set was simply eleven practice items of some sort, ordered a, b, c, d...k. But as soon as the student began to make errors (the items in boxes a and c), the item set changed, resulting in a set that is completely customized to accommodate student performance. The tutor--teacher or computer--would have to remember where in the item set to put the additional instances of items a and c, as indicated by the arrows in the figure.

Insert Figure 4 about here

Revising item sets is essentially a way in which lessons can be thought of as "intelligent," in that the program for the lesson literally redesigns itself while it is running, based solely on characteristics of the learner revealed to the program through each instance of the learner's actual performance. Good human teachers have the flexibility and good sense to make such on-the-spot modifications when student performance warrants it. There is nothing to prevent human instructional designers and programmers from telling the computer to do the same thing, to do it every time it is warranted, and to do it based upon individual performance, rather than upon group performance.

The model provides for revising the item set based upon any type of error made on any single item, with the exception of response convention errors.

Provide feedback for incorrect response. Superficially, this feature appears to be similar to feedback for correct responses. At the simplest level, knowledge of response (KOR) feedback can be provided for wrong responses. But, while (KOR) feedback might be adequate for correct responses to a wide variety of tasks, it is quite difficult to envision an instructional situation in which KOR feedback alone makes sense. Yet, some CAI relies heavily on "No. Try again" or "No. The answer is _____" as feedback for wrong responses.

The corollary to reinforcement feedback for correct responses is punishment for wrong responses, such as, "Sorry, stupid." With the possible exception of clearly using such feedback tongue-in-cheek in some games, we see no use for such feedback in CAI or elsewhere.

A type of feedback uniquely designed for wrong responses, by definition, is corrective feedback, in which the student is required to produce a correct response before moving on in the lesson. When a response is being taught (rather than, or in addition to, a discrimination), requiring the student to produce the desired response before advancing is appropriate. However, if discriminations alone are being taught, producing a correct response before advancing in the lesson may be characterized as little more than rat behavior.

Imagine, for instance, a multiple-choice vocabulary task that requires students to select a word from several choices that most closely matches

the target word in meaning. The student makes the wrong selection, is told which selection is correct, and then is required to select that correct choice before continuing. In this case, we are talking about only one keypress, but a completely wasted keypress. No discrimination and no retention is required for such a task, whose goal ostensibly is to achieve retained discrimination.

A similar situation is illustrated by this sequence:

1. A student types 9 for 3 + 7, then presses RETURN.

2. The feedback says, "No. Try 10."

3. The student presses RETURN, the feedback erases, and the student is confronted with exactly the same problem in exactly the same form.

In this case, the student is not simply copying a correct response from feedback (since the feedback was erased). But, on the other hand, the student need remember the information in the feedback for only a moment. Correct performance on the second trial is indicative of little more than the shortest of short-term retention. It is not evidence that anything was learned. It is, in short, another form of rat behavior, and an instance of getting the right answer for the wrong reason.

We suspect that corrective feedback is used in these inappropriate fashions by default. If the lesson does not revise the practice set (which is usually the case), then what else is the lesson designer to have the student do? Note how revising the set solves this problem. In the case of the multiple-choice example, that item (or one of the same type) is guaranteed to return in the practice set at a later time. Even though the task might only involve a simple verbal association, the order of choices can be changed in the reappearances, as well as one or more of the distractors, thereby requiring discrimination anew when the task reappears. Should it happen that a student cannot ever remember the association (in spite of being shown it at every error occurrence), the model allows for setting a limit on the number of incorrect trials permitted before routing the student elsewhere.

For practice items that are made up of a series of steps in a sequence, immediate repetition of the item <u>is</u> appropriate, because a time delay occurs between the feedback for the error and the repetition of the step on which the error was made. This time delay is the result of intervening steps in the sequence.

One type of feedback for wrong responses that is appropriate for some tasks might be characterized simply as <u>informational</u> feedback. When objectives include not only accuracy but quick recall, the feedback for an accurate, but slow, response cannot be expected to accomplish more than inform the student of the time that the response took. Such feedback may be made reinforcing, however, by indicating that the student is making improvements in response time.

The general category of feedback that is highly appropriate for most wrong response situations is instructional feedback. The degree of elaborateness for instructional feedback may vary considerably, but is determined by the nature of the task and the error that was made. Instructional feedback, which also may be characterized as learning guidance feedback, should be an adaptation of the learning guidance that was provided during the original modeling of the new skill. The rationale for this is simple: whatever learning guidance has proven most salient for an initial teaching presentation should also be most salient in response to a specific problem. The difference is that the guidance in the initial presentation needs to exhaust the possible applications of the new skill as generally as possible, because at that point in the lesson, the designer has no empirical knowledge of which aspect of the new skill will trouble any particular student. When used as feedback, however, the learning guidance can (and should be) narrowed to address the specific problems demonstrated by specific students.

During original modeling, learning guidance cannot easily be directed to learner differences, because those learner differences that are important to the instruction at hand are not revealed until students begin to respond in the lesson. It is at the point of responding that learning guidance should be tailored to learner differences: the guidance is modified to correct whatever difficulty the student may be having. This means that the greatest amount of learning guidance and the guidance that is of most value to a particular student are likely to occur during feedback, not during the modeling phase of the lesson.

Say, for example, that the student is to mark this sentence so that it is capitalized correctly:

The trip to san diego was a disaster for Professor smith.

Initial learning guidance for this item should have included rules and examples of their application for capitalizing the names of cities, people's names, people's titles, and the first word in a sentence. The feedback, however, need be no more (nor less) complex than the performance of the student warrants. Assume Student A produces this response:

The trip to San Diego was a disaster for Professor smith.

Although several capitalization rules may have been presented in initial presentations prior to working on this item, the student has only demonstrated problems with one of them: capitalizing a person's name. First of all, feedback for a correct response should be provided, since Student A did correctly capitalize San Diego. The only thing wrong with the response was the failure to capitalize "smith." We could limit the learning guidance to address only the rule for capitalizing the names of people, but we don't know why the error occurred. Did the student forget the rule, or not know that Smith is the name of a person, or confuse this rule with another? Or was the student simply not attending to the task? Because we don't know why the error occurred, we should

attempt to provide feedback that will account for all these possibilities. First, we can simply give the student a list of the rules presented thus far, giving this student the opportunity to compare rules or to recall the rule in question. If the student is able then to get the item correct, fine. If not, more specific feedback is in order, such as restatement of the rule in question and an indication that "Smith" is a person's name. These steps, combined with a revision of the item set, constitute specific instructional feedback.

Although the feedback described above for Student A may seem a bit elaborate relative to some sorts of feedback, as instruction it is relatively simple.

Consider this response by Student B:

the trip to san Diego was a Disaster for professor smith.

In this response, the student: (1) has incorrectly "uncapitalized" two words that were correct in the first place, but for different reasons; (2) has capitalized only one of the two parts of the city name; and (3) has capitalized a word that does not require capitalization. Instructional feedback that addresses all these difficulties will be elaborate, but appropriate, and will be based upon the learning guidance given originally.

In these examples, both Student A and Student B are given learning guidance that is keyed exactly to the performance of each, even though their performances on precisely the same item varied considerably. Computers are capable of delivering tasks such as this which approximate the "real world" application of many skills very closely, and of delivering a wide variety of feedback on such tasks when such feedback is indicated by a wide variety of student performance. Good teachers do this all the time. The extent to which courseware exploits these capabilities of computers might be one of the criteria used for judging the quality of courseware.

Instructional feedback may be even more elaborate than that which we have described. A subcategory of instructional feedback often used by teachers is what we call "nested feedback": essentially, a feedback routine that elicits more responses and gives additional feedback, all as part of the immediate feedback for a particular item. Let us assume that a student who has just been introduced to the spelling rule for doubling final consonants makes this error while trying to spell "tripped": "triped." A nested feedback routine could include the following additional questions, each with its own appropriate instructional feedback:

1. Does "trip" end with a consonant-vowel-consonant pattern?

2. Does the suffix begin with a vowel letter?

3. So do you have to double?

4. Which letter do you double?

5. So trip + ed = tripped.

When a routine of steps is required for the performance of a skill, the feedback should involve an interactive review of all the steps involved, which are the conditions for nested feedback.

In short, instructional feedback is based upon (1) the type of error a particular student makes, (2) the range of possible reasons for making that error (of those under the designer's control), and (3) the nature of the learning guidance provided in the initial teaching presentation.

After feedback is given for either correct or wrong responses, another response on another item from the practice set is elicited from the student.

Insert Figure 5 about here

Figure 5 illustrates the features of the model that have been discussed thus far in Section II. The most important aspect of the features in this part of the model is that each feature is expressed and activated upon the basis of students' performance with individual practice items. Although somewhat speculative, it is probably accurate to say that learning really takes place in this portion of the model, where:

1. the most precise, individualized learning guidance is provided, via performance-specific, instructional feedback.

2. formative assessment is continuous.

3. attention and retention are both provided for.

4. the lesson continuously redesigns itself, based solely upon the performance of individual students.

5. the actions of the computer are most "competitive" with the actions of a good teacher delivering one-to-one instruction.

Exhaust item set. In this model, we know the size of a given item set only for those students who perform flawlessly. For such students, the item set is frequently exhausted by responding correctly to every item once. In that case, if there are ten practice items in the set, the student exhausts the set by correctly responding to all ten items with no intervening errors. This, however, is not always the case.

Sometimes, a correct response on a particular practice item is not convincing evidence that the student has really learned what is being taught. The response conventions for the items, for instance, may be such that a guessing strategy has a fair chance of producing correct responses.

The chances of students getting correct responses for the wrong reason--guessing--are relatively high with choice-response tasks, those involving yes/no, true/false, and multiple choice. One method for diminishing the possible effect of guessing in these types of tasks (assuming they are appropriate in the first place) is to increase the "retirement criterion" for each item from one correct response to two, three, or more. If a practice set has ten items and a retirement criterion of two, the student who performs flawlessly must respond correctly to twenty tasks before exhausting the item set.

What we do not know is the size of the practice set for students who make errors. Each student who makes errors will receive different amounts and types of practice, depending on the nature and extent of the errors made. In effect, it is the student who controls the size of the practice set beyond the minimal "perfect performance" set initially specified for the lesson.

We would like to reiterate a most critical consideration at this point: students who make errors in the practice set--even a large number of errors--achieve the same level of mastery as students who perform flawlessly. That is, students who make errors must not only get items correct before those items are retired from the practice set, but must also get those items correct for the right reasons; that is, they must do so when time and other items intervene between the error and the correct response, and they cannot do so purely on the basis of short-term memory. Such performance is insured by the increasing ratio revision of the item set. Exhausting a practice set, therefore, is a powerful indication of mastery, prior to any formal evaluation of mastery.

When the item set is exhausted, ending the current practice session, several options are possible. The student may be branched back to the framing stage of new, but related, instruction, or to a new set of application items. One of these alternatives is common for many types of lessons. Once adequate instruction for achieving the objectives of the lesson has been given, the student is branched to formal evaluation.

Within any given practice session, it may be determined that a student needs to be branched out of a practice set early. This step is taken when a student is making an inordinate number of errors or particularly severe errors. That student may be automatically sent to earlier instruction, or back to the beginning of the lesson (at the framing stage), or if necessary, the lesson can indicate that this student requires teacher intervention. These options are indicated in Figure 6.

Insert Figure 6 about here

Section III

Immediate posttest. This is a curriculum-based evaluation of the skill or skills taught in the model, or more likely, in a group of passes through the model. Typically, poor performance on end-of-lesson evaluations is indicative of the need to revise the lesson (Dick, 1981). This value of posttests is retained in the model. But because of continuous assessment and the continuous revision of item sets and learning guidance, this role of the posttest is diminished relative to the affirmative value of the posttest to indicate that what was taught was also learned. Revisions, no doubt, will be implied anyway. But the in-lesson, continuous assessment ordinarily means that less drastic and fewer revisions will be implied. Given that the theory and the reality of doing revisions often conflict irresolvably, this feature of continuous assessment/revision within lessons is potentially of great value.

The fact that the tutorial model requires mastery of the item set also diminishes the value of an immediate posttest to some extent (but not delayed posttests), given that downward exit from the item set is indicative of mastery quite similar to that tested for by an immediate posttest. However, a feature of formative evaluations is that only limited feedback is given for each item, usually restricted to an acknowledgment that the input was recorded. The immediate posttest, therefore, is a "cleaner" test than the continuous assessment of the item set, in which the feedback for one item might very well (and desirably) serve as a prompt for correct responding on other items.

The items for the posttest can be identical to items used in the lesson only when generalizations are not being taught. If the objective of the lesson is for students to learn a set of associations, such as chemical names and their symbols, then it makes little sense for the posttest to test on something more than, less than, or different from what was taught in the lesson.

For tests of generalization, however, the items cannot be taken from the instruction and still be considered tests of the generalization (Engelmann & Carnine, 1982; Merrill & Tennyson, 1977; Tiemann & Markle, 1978). On the other hand, the items must represent the the same range of the generalization that was taught in the lesson, and must use same response modes (unless provision was specifically made within the lesson for generalization to new response modes).

Another way of viewing posttests is that they are the most concrete representation possible of lesson objectives. If the set of posttest items, performance criteria on those items, and the relationship of those items to instruction do not correspond to lesson objectives, then either we are testing something we were not interested in teaching, or the statement of objectives does not represent our actual goals for the lesson. Given the need for a high level of correspondence between objectives and posttests, the development of a posttest, along with

performance criteria and an indication of the relationship between testing and teaching items, may be considered an alternative to the written description of objectives. The two methods of expressing objectives are redundant, which simply means that either both methods can be used (and subsequently can provide a mechanism for double-checking one against the other), or that only one method is minimally necessary (in which case, the one method should be the test, since it is otherwise a necessary component of the instruction).

Delayed posttest. The form of this test exactly parallels that of the immediate posttest. The delayed posttest, however, is the more meaningful of the two, because it more closely approximates the "real" application of new skills, since it takes place after time has passed since the initial learning. The length of delay must be determined by an analysis of the likely applications of the skill. If the instruction aspires to teach "lifetime" skills, then sets of delayed tests ranging from "short" delays to delays of weeks or months are appropriate. Skills that are envisioned primarily as facilitators for upcoming skills should be delay-posttested at a time that shortly precedes instruction in an upcoming skill. (When sets of related lessons are sequenced as a curriculum, the pretest for one lesson may well serve the function of a delayed posttest for an earlier lesson in the sequence.)

When discrimination performance is high on an immediate posttest but low on a parallel delayed posttest, lack of practice adequate for retaining the skill over the tested period is clearly indicated. This implies that the student should be sent back to the phase of instruction covering such skills, or that the instruction should be revised, or both. Students who come back up to performance criteria will exit the model relatively quickly, while those who need more practice with the skill will get it.

The method of selecting items for both the immediate and delayed posttests is nearly identical to that for selecting practice items: ask, if the student performs without making a single error, how many items and what kind of items would I accept as indicative that the student has learned what I intended?

Figure 7 shows the complete model.

Insert Figure 7 about here

Applications of the Model

Lesson Design

Although some topics can be covered exhaustively in one "pass" through the model, that situation is more the exception than the rule. Typically, several passes are required. The following is an outline of two lessons from the University of Illinois PLATO Curriculum Project (PCP) Language Arts Curriculum, which deal with the topic of capitalization:

Lesson 1

Session 1 - Capitalizing days, months, holidays, and the first word in a sentence.

Session 2 - Capitalizing cities, states, and countries.

Session 3 - Consolidation.

Lesson 2

Session 1 - Capitalizing the names of people and their titles, words related to the names of countries, the word "I," names of streets, and the first word in a direct quotation.

Session 2 - Consolidation

Session 3 - Capitalizing titles.

Session 4 - Consolidation.

Each session in the outline represents one kind of pass or another through the model. The consolidation sessions consist of passes through the "practice set" features of the model only. Each such session provides practice applying capitalization rules presented in all preceding sessions. All other sessions consist of passes through the entire model from framing through the practice set features, though the specifics of each session vary. The seven sessions of the two lessons, therefore, represent seven passes through the model.

This sequence of lessons by no means exhausts the topic, capitalization. (Further capitalization lessons are currently under development at PCP.) It does serve to indicate, however, one kind of relationship between the tutorial model and instructional lessons.

The nature of the model raises an important practical question: What range of skills can this tutorial model realistically be expected to cover?

Obviously, tutorials in general and our model in particular are incapable of accommodating all the goals of education. With respect to the most ambitious educational goals, we share Gagné's sentiment: We do not know how to produce an Albert Einstein or a Leonardo da Vinci. The conditions for achieving the highest levels of inventiveness, aesthetic creativity, and intellect are not apparent to us, beyond our belief that such desirable skills are dependent to some extent upon mastery of more manageable skills, such as simple concept acquisition, rule application, and even verbal knowledge.

The model, however, seems capable of accommodating a rather wide range of skills typically taught in schools, particularly when sequences of lessons are considered. For some skills, a single pass through the model appears adequate. For instance, if the skill to be learned is applying a formula for calculating the area of a triangle, a single pass ought to be adequate, whether the rule is presented directly or is "discovered" through application opportunities and feedback. (We don't advocate the isolated development of lessons such as this, but wish only to indicate that the model can accommodate the complete learning of such a skill.)

But for many skills, such as identifying the main idea of a variety of passage types, a complex sequence of model implementations is implied, each dealing with one of the component skills prerequisite for identifying main ideas, skills related to inference, anaphora, classification, and so on. The number and sequence of model applications are dependent upon the results of the task analysis that is performed for the skill.

By interfacing the tutorial model with other categories of computer lessons, an even larger skill range can be addressed. Students can be routed between simulations and the model, for example, on the basis of performance. One strategy for such interaction might proceed as follows: Allow students initially to enter the simulation, and keep track of the extent to which each student is achieving the instructional goal of the simulation. For those students who are not achieving the goal (either in terms of effectiveness or efficiency), provide for routing to the tutorial, which has the capability of delivering variable practice and specific feedback to those students who need it, and which is directed to the specific subskills from the simulation with which a student is having difficulty.

The interface of the tutorial with games can be accomplished in the same manner, or can be more fully integrated by simply imposing game features over the interactive features of the tutorial, features such as the accumulation of points, racing a clock, highest score records, and the like. Such strategies for games and simulations assume, of course, that there are instructional goals for these categories that incorporate efficiency and effectiveness.

We use two data-keeping features in our lessons that are not readily visible in the model. In actual classroom practice, it is often the

case that a student's work on the computer has to stop at a time that does not correspond to an ending point in a lesson index or menu. When that student enters the lesson at a later time, he or she must unnecessarily repeat some material. This situation can be particularly wasteful if the lesson produces custom-made practice sets, since those sets will be lost. To prevent this loss of student time and computer resource, we save the custom practice set and where the student was in that set, allowing each student to continue at a later time where he or she left off, even when that location was between segments listed in the lesson index.

Another type of data that can be kept is the most difficult items in a given practice set, defined empirically by a student's performance in that practice set. Although practice sets are usually developed according to the "perfect performance" criteria discussed earlier, the "most difficult item" data can be used to create custom-made practice sets subsequent to the first one, before the student ever enters that practice set. For instance, Session 1 of a given lesson could be made up initially of fifteen practice items, A-O, with items for Session 2 unspecified. Then if items B, C, F-H, K, and N proved to be the most difficult items for Session 1, Session 2 would be custom designed by the lesson itself for that particular student. Session 2 would be composed of that student's most difficult items and possibly items selected according to other criteria.

Research

Another value of a model such as this may be its use in experimental research. Although numerous studies are done on individual features of instruction, most instructional design models, including this one, comprise features that are assumed to work interactively. That is, the value claimed for any single feature of a model is contingent upon that feature's interaction with all the other features of the model. For that reason, research on individual features is of somewhat limited value. For example, several studies have been conducted on immediate versus delayed feedback--essentially, an isolated feature of any design model. The value of such research for instructional design is limited, because the designer must implement several features simultaneously, and therefore needs empirical support for the interactions of those several features.

Empirical information on the effectiveness, efficiency, and viability of a design model is contingent upon the existence of several models, each explicated sufficiently for ready implementation and comparison. It is to this end primarily that we have offered this model.

Final Considerations

We have indicated elsewhere that tutorial models in general and ours in particular are incapable of addressing all the goals of education, but still have the potential for accommodating literally hundreds and hundreds of instructional goals. Models, however, cannot tell us what those instructional goals should be. That is and remains an essentially

subjective judgment, made hopefully by those sensitive to the needs of students through their school careers and lifetimes. Neither can models contribute to the difficult design job of task analysis, upon which so many features of models are dependent. No model, in short, has much practical value to those who have not established clear, relevant instructional goals, or who are unwilling or unable to conduct a meaningful task analysis. A commitment to goal formulation and task analysis is prerequisite to a commitment to designing any instruction in accordance with instructional design theory, computer-based or otherwise.

We have indicated repeatedly that a requirement for producing theory-based computer tutorials is expert programming. For most stand-alone systems, this usually means utilizing programmers who are proficient with assembly languages, as well as the standard (and some not so standard) languages. It is much easier, however, to implement any sophisticated design model with a utility package designed specifically to enhance the power of a standard language. EnBASIC (1983) is such a utility. EnBASIC allows users to make sophisticated judgments of student responses, enables the use of a variety of character sets, facilitates the use of clean animations, and has a host of other features, all of which would normally require assembly-language programming.

We have also referred frequently to comparisons between computer-delivered tutorials and one-to-one, teacher-directed tutorials. When the relative advantages and disadvantages of each are summed up, there is no doubt that teacher-directed tutorials have the greater potential for delivering effective and efficient, theory-based instruction. The key to this comparison, however, is that both modes of instruction require delivery to one student at a time. That constraint, while common (and mandatory for stand-alone systems) for computer-based instruction, is a prohibitively expensive one to impose in most teacher-directed situations.

Almost every advantage the teacher has over the computer in one-to-one instruction is greatly diluted or eliminated entirely once we begin to expand the range of a teacher's responsibility to more than one student at a time. Two students, for instance, who make different errors at the same time on a given practice item simply cannot receive individualized instruction from a human teacher. Either the teacher corrects each error--at the cost of efficienty--or the teacher gives one general correction for both errors--at the cost of effectiveness. The implication of this is not, of course, that computer-based instruction can or should replace teachers, but that computer-based instruction can go a long way toward freeing teachers to do more one-to-one, highly individualized, adaptive, response-sensitive instruction.

References

Blankenship, C., & Lilly, S. Mainstreaming Students with Learning and Behavior Problems: Techniques for the Classroom Teacher. New York: Holt, Rinehart, & Winston, 1981.

Block, J. H. (Ed.). Mastery Learning: Theory and Practice. New York: Holt, Rinehart & Winston, 1971.

Bloom, B. S. (Ed.). Taxonomy of Educational Objectives: Handbook I: Cognitive Domain. New York: Longman, Inc., 1956.

Bork, A. Educational Technology and the Future. Journal of Educational Technology Systems, 1981-82, 10, 9-21.

Dick, W. Instructional Design Models: Future Trends and Issues. Educational Technology, July 1981, 29-32.

Dugdale, S., Kibbey, D., & Cohen, B. Darts. Urbana, Il.: Board of Trustees of the University of Illinios, 1978. (Computer program)

Engelmann, S. & Carnine, D. Theory of Instruction: Principles and Applications. New York: Irvington Publishers, Inc., 1982.

Fisher, F. D., Computer-Assisted Instruction: What's Not Happening? Journal of Computer-based Instruction, Summer 1982, 9, 19-27.

Friend, J. & Milojkovic, J. D. Successful Student/ Microcomputer Interactions. The Apple Journal, 1982, 1, 3-7.

Gagné, R. M. The Conditions of Learning (2nd ed.). New York: Holt, Rinehart, & Winston, 1970.

Markle, S. M. Designs for Instructional Designers. Champaign, Il.: Stipes Publishing Company, 1978.

Merrill, M. D. & Tennyson, R. D. Teaching Concepts: An Instructional Design Guide. Engelwood Cliffs, N.J.: Educational Technology Publications, 1977.

Mitzel, H. E. On the Importance of Theory in Applying Technology to Education. Journal of Computer-based Instruction, May 1981, 7, 93-98.

Park, O. & Tennyson, R. D. Adaptive Design Strategies for Selecting Number and Presentation Order of Examples in Coordinate Concept Acquisition. Journal of Educational Psychology, 1980, 72. 362-370.

Rosenshine, B. Teaching Functions in Instructional Programs. Paper prepared for the National Institute of Education's Conference on the implications of Research on Teaching for Practice, Airlie House, Virginia, February 1982.

Siegel, M. A. & Misselt, A. L. An Adaptive Feedback and Review Paradigm for Computer-based Drills. (CERL Report E-25). Urbana, Il.: Computer-based Education Research Laboratory, University of Illinios, February, 1983.

Taylor, R. T. (Ed.). The Computer in the School: Tutor, Tool, Tutee. New York: Teachers College Press, 1980.

Tenzcar, P., Smith, S., & Avner, A. EnBASIC Authoring System (2nd ed.). Wentworth, N.H.: COMPress Division of Van Nostrand Reinhold, 1983. (Computer program and text)

Tiemann, P. W. & Markle, S. M. Analyzing Instructional Content: A Guide to Instruction and Evaluation. Champaign, Il.: Stipes Publishing Company, 1978.

Wager, W. Design Considerations for Instructional Computing Programs. Journal of Educational Technology Systems, 1981-82, 10, 261-270.

Footnote

[1]Although pretesting is the first feature of the model that a student would encounter, it is one of the last elements of the lesson to be developed because the pretest is curriculum-based: the curriculum on which the pretest is based must be developed before the pretest. Details on developing curriculum-based assessment are available in Mainstreaming Students with Learning and Behavior Problems, by Blankenship and Lilly (1981). The assessment tools explicated by Blankenship and Lilly are clearly appropriate for use with students other than those with specific learning or behavior problems. It is our opinion, in fact, that the applications of techiniques in this book are far greater than mainstreaming, learning problems, and behavior problems.

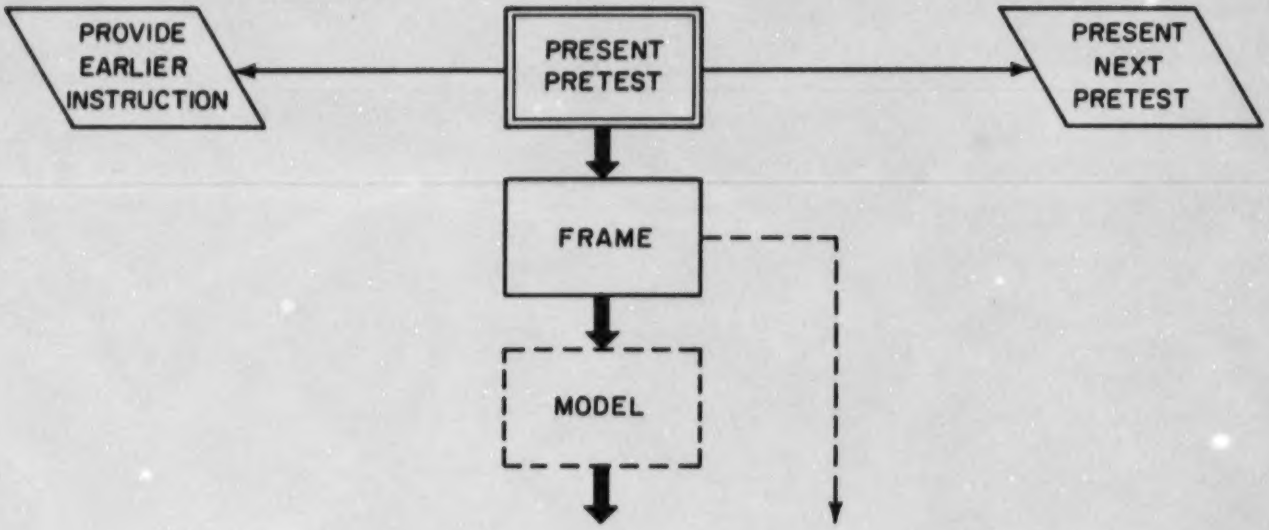

Figure 1. First part of the tutorial model.

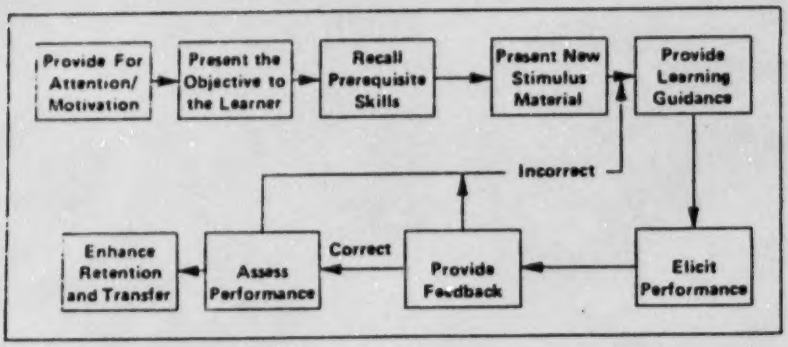

Figure 2. Wager model.

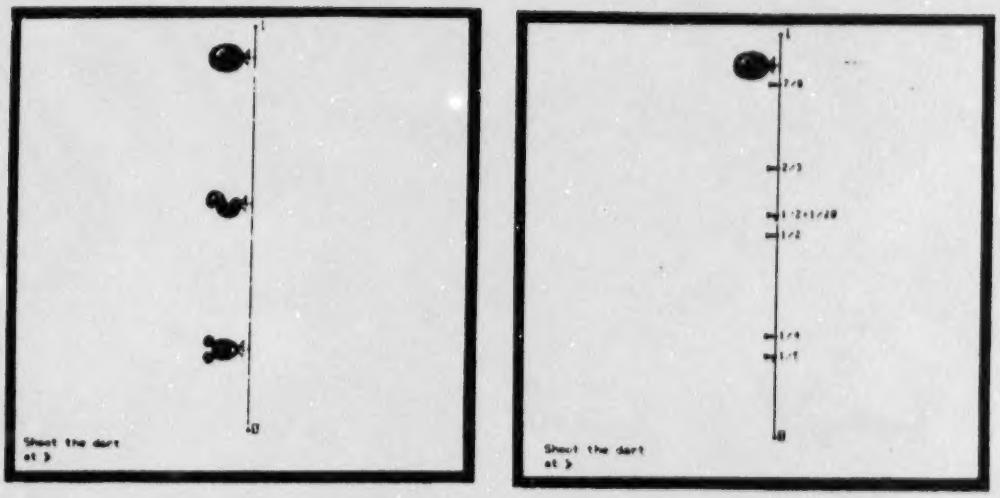

Figure 3. Darts.

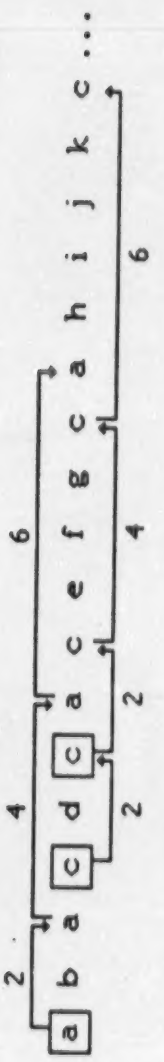

Figure 4. Increasing ratio review (from the CFP).

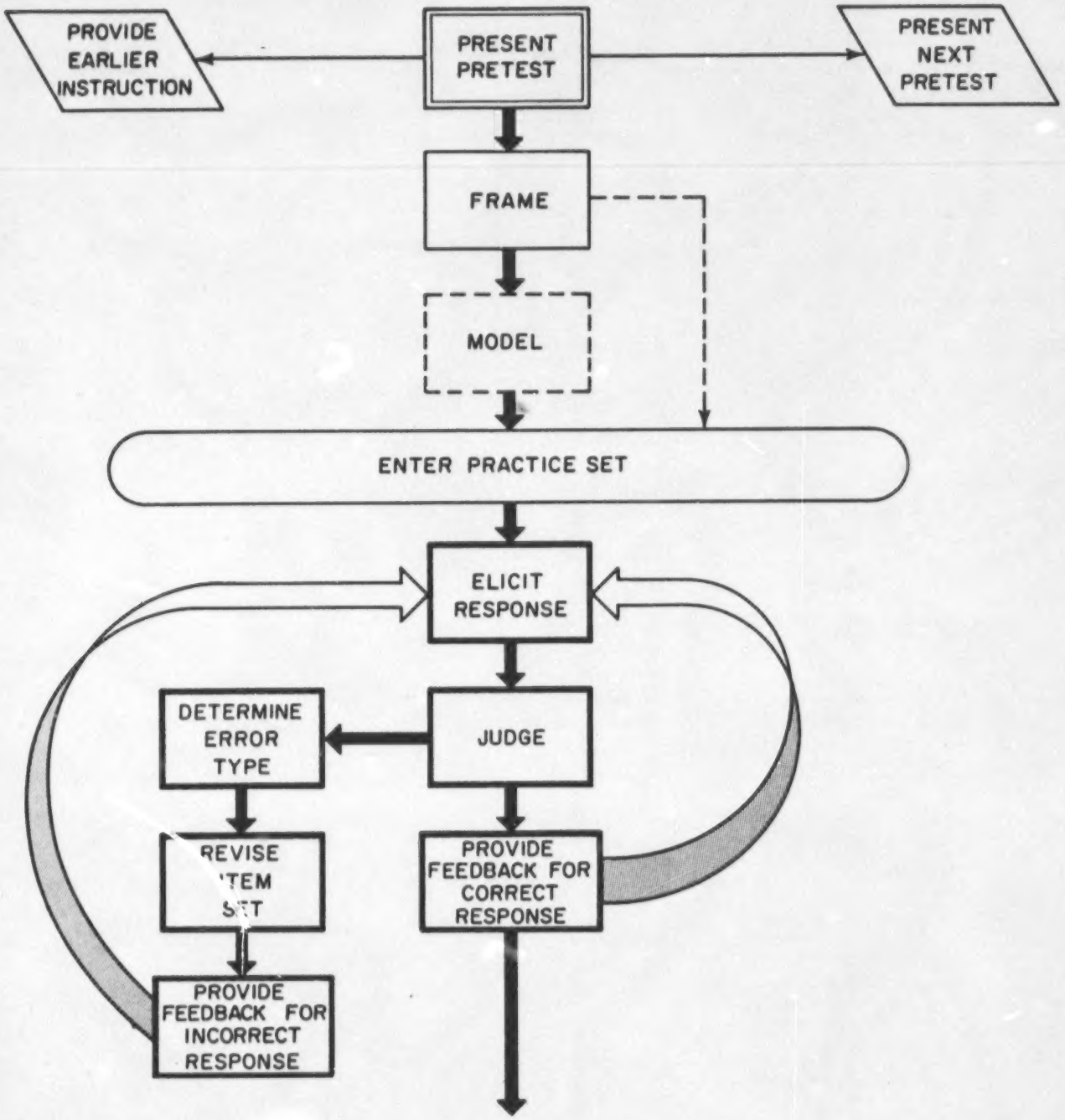

Figure 5. Second part of the tutorial model.

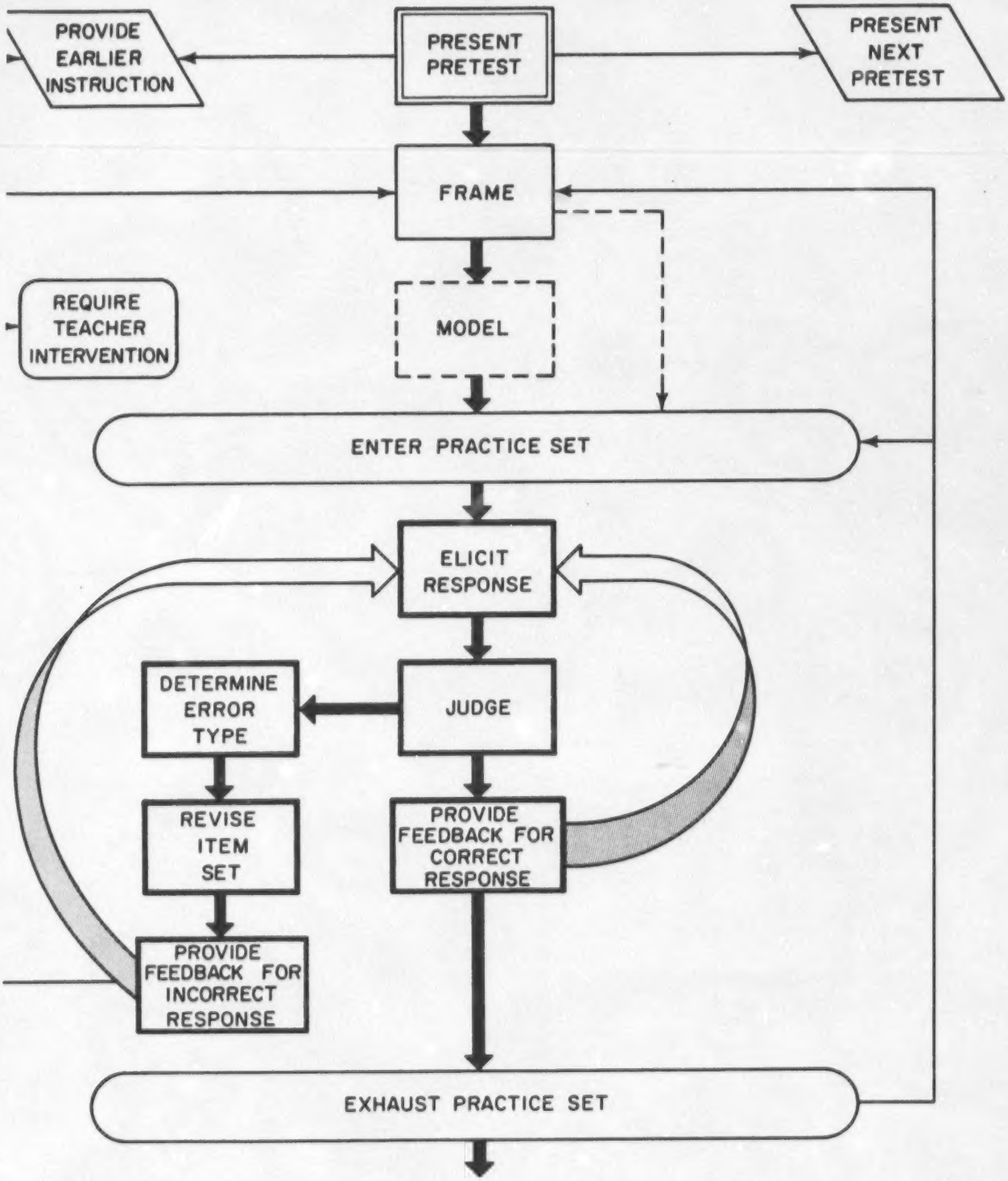

Figure 6. Third part of the tutorial model.

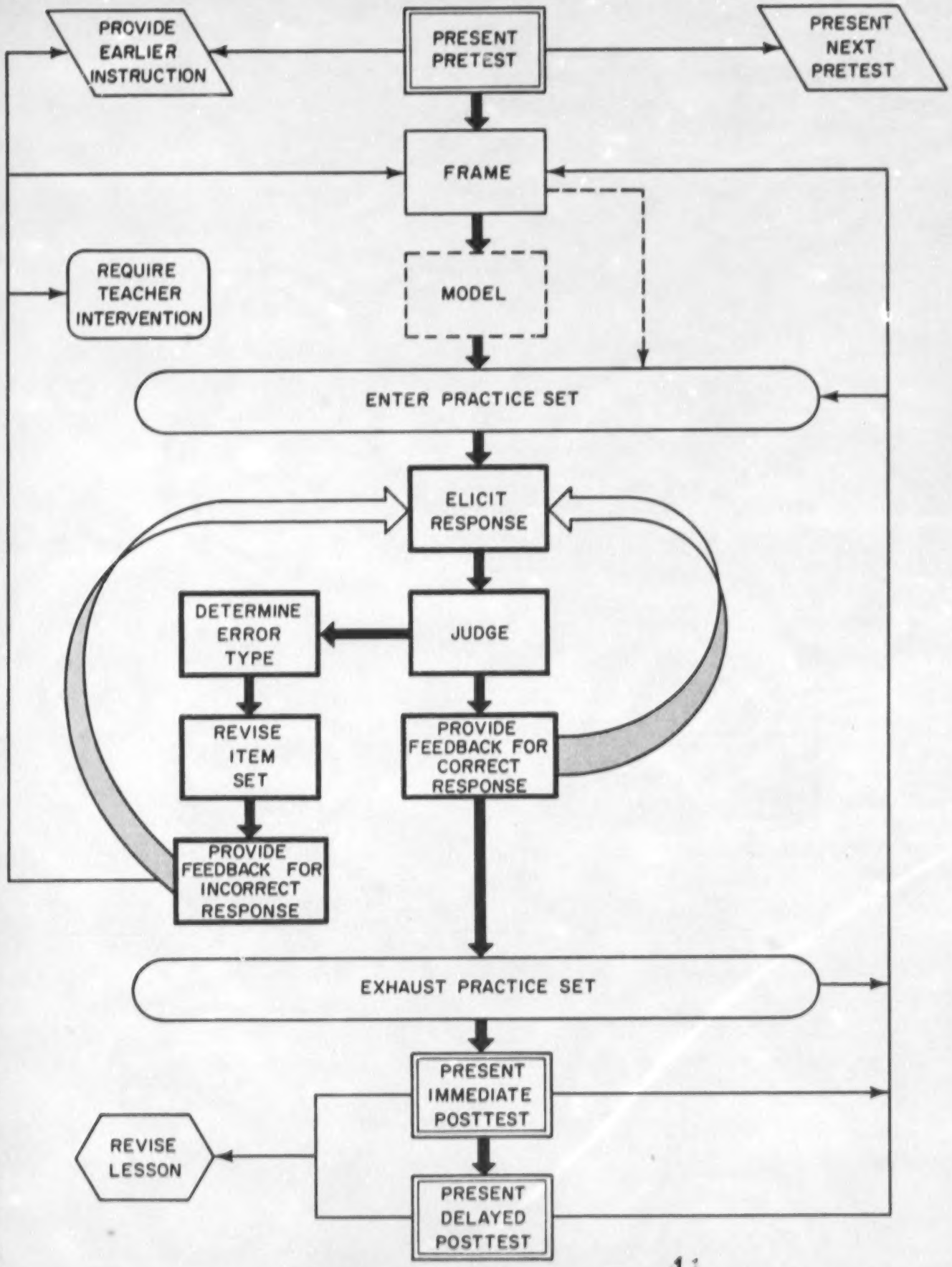

Figure 7. The complete tutorial model.